Filtrated Spectral Algebraic Subspace Clustering [*]

Manolis C. Tsakiris and René Vidal
Center for Imaging Science, Johns Hopkins University
3400 N. Charles Street, Baltimore, MD, 21218, USA
m.tsakiris,rvidal@jhu.edu

Abstract

Algebraic Subspace Clustering (ASC) is a simple and elegant method based on polynomial fitting and differentiation for clustering noiseless data drawn from an arbitrary union of subspaces. In practice, however, ASC is limited to equi-dimensional subspaces because the estimation of the subspace dimension via algebraic methods is sensitive to noise. This paper proposes a new ASC algorithm that can handle noisy data drawn from subspaces of arbitrary dimensions. The key ideas are (1) to construct, at each point, a decreasing sequence of subspaces containing the subspace passing through that point; (2) to use the distances from any other point to each subspace in the sequence to construct a subspace clustering affinity, which is superior to alternative affinities both in theory and in practice. Experiments on the Hopkins 155 dataset demonstrate the superiority of the proposed method with respect to sparse and low rank subspace clustering methods.

1. Introduction

Subspace clustering is the problem of clustering a collection of points drawn approximately from a union of linear subspaces. This is an important problem in pattern recognition with diverse applications from computer vision [22] to genomics [15].

Related Work. Early subspace clustering methods were based on alternating between finding the subspaces given the clustering and vice versa [2, 21, 17], and were very sensitive to initialization. The need for good initialization motivated the development of an algebraic technique called Generalized Principal Component Analysis (GPCA) [26], which solves the problem in closed form. The key idea behind GPCA is that a union of n subspaces can be represented by a collection of polynomials of degree n, with the property that their gradients at a data point give the normals to the subspace passing through that point. This

is exploited in [24] and [8] for clustering a known number of subspaces. The recent Abstract Algebraic Subspace Clustering (AASC) method of [19, 20], unifies the ideas of [8, 26], into a provably correct method for the decomposition of a union of subspaces to its constituent subspaces. However, while in theory GPCA and AASC are applicable to subspaces of any dimensions, in practice the estimation of the subspaces is sensitive to data corruptions.

The need for methods that can handle high-dimensional data corrupted by noise and outliers motivated the quest for better subspace clustering affinities. State-of-the-art methods, such as *Sparse Subspace Clustering* [4, 5, 6] and *Low Rank Subspace Clustering* [12, 7, 23, 11], exploit the fact that a point in a union of subspaces can always be expressed as a linear combination of other points in the subspaces. Sparse and low rank representation techniques are then used to compute the coefficients, which are then used to build a subspace clustering affinity. These methods perform very well when the subspace dimensions are much smaller than the dimension of the ambient space, the subspaces are sufficiently separated and the data are well distributed inside the subspaces [6, 16]. However, these methods fail when the dimensions of the subspaces are large, e.g. a union of hyperplanes, which is the case where GPCA, to be henceforth called Algebraic Subspace Clustering (ASC), performs best. In addition, sparse methods produce low Inter-Class Connectivity, but the Intra-Class Connectivity is also low due to sparsity, leading to over-segmentation issues. Conversely, Low-Rank and ℓ_2 methods produce high Intra-Class Connectivity (since they are less sparse) but this also leads to high Inter-Class Conectivity. Consequently, there is a strong need for developing methods that produce high Intra-Class and low Inter-Class connectivity.

Paper Contributions. The main contribution of this paper is to propose a new subspace clustering algorithm that can handle noisy data drawn from a union of subspaces of different dimensions. The key idea is to construct for each data point (the *reference point*) a sequence of projections onto hyperplanes that contain the *reference subspace* (the subspace associated to the reference point). The norms of the

---

projected data points are used to define their affinity with the reference point. This process leads to an affinity matrix of high intra-class and low cross-class connectivity, upon which spectral clustering is applied. We provide a theorem of correctness of the proposed algorithm in the absence of noise as well as a variation suitable for noisy data. As a secondary contribution, we propose to replace the angle-based affinity proposed in [26] by a superior distance-based affinity. This modification is motivated by the fact that the angle-based affinity is theoretically correct only in the case of hyperplanes, and is not a good affinity for subspaces of varying dimensions. Our experiments demonstrate that the proposed method outperforms other subspace clustering algorithms on the Hopkins 155 motion segmentation database as well as on synthetic experiments for arbitrary-dimensional subspaces of a low-dimensional ambient space.

2. Algebraic Subspace Clustering: A Review

We begin with a brief overview of the ASC theory and algorithms. We refer the reader to [25, 26, 3, 14] for details.

Subspace Clustering Problem. Let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be a set of points that lie in an unknown union of $n > 1$ subspaces $\mathcal{A} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n$, where $\mathcal{S}_i$ a linear subspace of $\mathbb{R}^D$ of dimension $d_i < D$. The goal of subspace clustering is to find the number of subspaces, a basis for each subspace, and cluster the data points based on their subspace membership, *i.e.*, find the correct decomposition or *clustering* of $\mathcal{X}$ as $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_n$, where $\mathcal{X}_i = \mathcal{X} \cap \mathcal{S}_i$. To make the subspace clustering problem well-defined, we need to make certain assumptions on the geometry of both the subspaces $\mathcal{S}_i$ and the data $\mathcal{X}$. In this work we assume that the underlying union of subspaces $\mathcal{A}$ is *transversal* [14], which in particular implies that there are no inclusions between subspaces. Moreover, we assume that $\mathcal{X}_i \cap \mathcal{S}_{i'} = \emptyset$ for $i \neq i'$, *i.e.*, each of the given points is associated to a unique subspace. This guarantees that the above decomposition of $\mathcal{X}$ is in fact a partition, and it is unique. A final assumption that we need is that the data $\mathcal{X}$ are rich enough and in general position (see Definition 1).

Unions of Subspaces as Algebraic Varieties. A key idea behind ASC is that a union of n subspaces $\mathcal{A} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n$ of $\mathbb{R}^D$ is the *zero set* of a finite set of homogeneous polynomials of degree n with real coefficients in D indeterminates $(x_1, \ldots, x_D)$. Such a set is called an *algebraic variety* [1]. For example, a union of n hyperplanes $\mathcal{A} = \mathcal{H}_1 \cup \cdots \cup \mathcal{H}_n$, where the ith hyperplane $\mathcal{H}_i = \{\boldsymbol{x} : \boldsymbol{b}_i^\top \boldsymbol{x} = 0\}$ is defined by its normal vector $\boldsymbol{b}_i \in \mathbb{R}^D$, is the zero set of

$$p(x) = (\boldsymbol{b}_1^\top x)(\boldsymbol{b}_2^\top x) \cdots (\boldsymbol{b}_n^\top x). \qquad (1)$$

Likewise, the union of a plane with normal $\boldsymbol{b}$ and a line with normals $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^3$ is the zero set of the two polynomials $p_1(x) = (\boldsymbol{b}^\top x)(\boldsymbol{b}_1^\top x)$ and $p_2(x) = (\boldsymbol{b}^\top x)(\boldsymbol{b}_2^\top x)$.

Observe that these *vanishing polynomials* are homogeneous of degree n, where n is the number of subspaces. Moreover, they are factorizable into linear forms, with each subspace contributing a linear form to the product. Each such linear form is in turn defined by a normal vector to the subspace.

Finding Vanishing Polynomials. Note that the coefficients of the polynomials associated with a union of subspaces $\mathcal{A}$ can be obtained from sufficiently many samples $\mathcal{X} \subset \mathcal{A}$ in *general position* by solving a linear system of equations.

Definition 1 *We say that the data $\mathcal{X} \subset \mathcal{A}$ is in general position if a degree n polynomial vanishes on $\mathcal{X}$ if and only if it vanishes on the underlying union of subspaces $\mathcal{A}$.*

For example, if $\mathcal{A}$ is a union of two planes in $\mathbb{R}^3$ with normals $\boldsymbol{b}_i = (b_{i1}, b_{i2}, b_{i3})$, $i = 1, 2$, then we can write p as

$$p(x) = (b_{11}x_1 + b_{12}x_2 + b_{13}x_3)(b_{21}x_1 + b_{22}x_2 + b_{23}x_3)$$
$$= c_1 x_1^2 + c_2 x_1 x_2 + \cdots + c_6 x_3^2 = \boldsymbol{c}^\top \nu_2(x), \qquad (2)$$

where $\boldsymbol{c} = (c_1, \ldots, c_6)$ and $\nu_2(x) = (x_1^2, x_1 x_2, \ldots, x_3^2)$. Thus, we can find the vector of coefficients $\boldsymbol{c}$ by solving the set of linear equations $\boldsymbol{c}^\top \nu_2(\boldsymbol{x}_j) = 0$ for $j = 1, \ldots, N$. More generally, each polynomial of degree n can be written as $p(x) = \boldsymbol{c}^\top \nu_n(x)$, where $\nu_n : \mathbb{R}^D \rightarrow \mathbb{R}^{\mathcal{M}_n(D)}$ is the Veronese embedding of degree n that maps a point $\boldsymbol{x} \in \mathbb{R}^D$ to all $\mathcal{M}_n(D) := \binom{n+D-1}{n}$ distinct monomials of degree n in the entries of $\boldsymbol{x}$. Consequently, a basis for the set of polynomials of degree n that vanishes in $\mathcal{X}$ can be found by computing a basis for the right nullspace of the embedded data matrix, *i.e.*, by solving the linear system:

$$\mathcal{V}_n(\mathcal{X})\boldsymbol{c} = [\nu_n(\boldsymbol{x}_1)\, \nu_n(\boldsymbol{x}_2)\, \cdots\, \nu_n(\boldsymbol{x}_N)]^\top \boldsymbol{c} = \boldsymbol{0}. \qquad (3)$$

However, the polynomials obtained by the above procedure may not factorize into a product of linear forms because the space of factorizable polynomials is not a linear space, e.g $(x_1 + x_2)x_1 - (x_1 - x_2)x_2 = x_1^2 + x_2^2$ is not factorizable.

Polynomial Differentiation Algorithm. Even though an elegant solution based on polynomial factorization exists for the case of hyperplanes [25], it has not been generalized for subspaces of different dimensions. However, an alternative solution has been achieved by observing that given *any* degree n vanishing polynomial p on $\mathcal{A}$, and a point $\boldsymbol{x}$ in $\mathcal{A}$, the gradient of p evaluated at $\boldsymbol{x}$ will be orthogonal to the subspace associated with point $\boldsymbol{x}$ (see [26] and [14] for a geometric and algebraic argument respectively). Consequently, for the purpose of computing normal vectors to the subspaces, it is enough to compute general vanishing polynomials of degree n. The set of all such polynomials, denoted $\mathcal{I}_{\mathcal{X},n}$, is a finite-dimensional vector space and a basis can be computed as a basis of the right nullspace of the Veronese matrix $\mathcal{V}_n(\mathcal{X}) := [\nu_n(\boldsymbol{x}_1)\, \nu_n(\boldsymbol{x}_2)\, \cdots\, \nu_n(\boldsymbol{x}_N)]^\top$, where $\nu_n : \mathbb{R}^D \rightarrow \mathbb{R}^{\mathcal{M}_n(D)}$ is the Veronese embedding

of degree n that maps a point $\boldsymbol{x} \in \mathbb{R}^D$ to all $\mathcal{M}_n(D) := \binom{n+D-1}{n}$ distinct monomials of degree n in the entries of $\boldsymbol{x}$. Having a basis $p_1, \ldots, p_s$ for $\mathcal{I}_{\mathcal{X},n}$, it can be shown that the subspace associated to a point $\boldsymbol{x} \in \mathcal{X}$ can be identified as the orthogonal complement of the subspace spanned by the vectors $\nabla p_1|_{\boldsymbol{x}}, \nabla p_2|_{\boldsymbol{x}}, \ldots, \nabla p_s|_{\boldsymbol{x}}$ [26, 14]. Then we can remove the points that lie in the same subspace as $\boldsymbol{x}$ and iterate the procedure with the remaining points until all subspaces have been identified. It is remarkable that this procedure is provably correct for a known number of subspaces of arbitrary dimensions. Even though this result is general and insightful, algorithms that are directly based on it are extremely sensitive to noise. The main reason is that any procedure for estimating the dimension of the nullspace will unavoidably involve thresholding the singular values of $\mathcal{V}_n(\mathcal{X})$, which will in turn yield very unstable estimates of the subspaces and subsequently poor clustering of the points.

Spectral Algebraic Subspace Clustering Algorithm. In the interest of enhancing the robustness of ASC in the presence of noise and obtaining a working algebraic algorithm, the standard practice has been to apply a variation of the polynomial differentiation algorithm based on spectral clustering. More specifically, given noisy data $\mathcal{X}$ lying close to a union of n subspaces $\mathcal{A}$, one computes an approximate vanishing polynomial p whose coefficients are given by the right singular vector of $V_n(\mathcal{X})$ corresponding to its smallest singular value. Given p, one computes the gradient of p at each point in $\mathcal{X}$ (which gives a normal vector associated with each point in $\mathcal{X}$), and builds an affinity matrix between points $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ as the cosine of the angle between their corresponding normal vectors, *i.e.*,

$$ \boldsymbol{C}_{jj'} = \Big| \Big\langle \frac{\nabla p|_{\boldsymbol{x}_j}}{||\nabla p|_{\boldsymbol{x}_j}||}, \frac{\nabla p|_{\boldsymbol{x}_{j'}}}{||\nabla p|_{\boldsymbol{x}_{j'}}||} \Big\rangle \Big|. \qquad (4) $$

This affinity is then used as input to any spectral clustering algorithm to obtain the clustering $\mathcal{X} = \cup_{i=1}^n \mathcal{X}_i$. We call this Spectral ASC method with *angle-based affinity* as SASC-A. To gain some intuition on $\boldsymbol{C}$, suppose $\mathcal{A}$ is a union of n hyperplanes and that there is no noise. Then p must be of the form in (1). In that case $\boldsymbol{C}_{jj'}$ is simply the cosine of the angle between the normals to the hyperplanes that are associated with points $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$. If both points lie in the same hyperplane, their normals must be equal, and hence $\boldsymbol{C}_{jj'} = 1$. Otherwise, $\boldsymbol{C}_{jj'} < 1$ is the cosine of the angles between the hyperplanes. Thus, assuming that these angles are not small, and that the points are well distributed on the union of the hyperplanes, spectral clustering on the affinity matrix $\boldsymbol{C}$ will in general yield the correct clustering. Even though SASC-A is much more robust in the presence of noise than purely algebraic methods for the case of a union of hyperplanes, it is fundamentally limited by the fact that it applies only to unions of hyperplanes. Indeed, if

the orthogonal complement of a subspace $\mathcal{S}$ has dimension greater than 1, there may be points $\boldsymbol{x}, \boldsymbol{x}'$ inside $\mathcal{S}$ such that the angle between $\nabla p|_{\boldsymbol{x}}$ and $\nabla p|_{\boldsymbol{x}'}$ is as large as $90°$. In such instances, points associated to the same subspace may be weakly connected and thus there is no guarantee for the success of spectral clustering.

Abstract Filtration Scheme. Motivated by the limitation of the polynomial differentiation algorithm to a known number of subspaces, and the association of undesired *ghost-subspaces* with the recursive method of [8], an alternative algebraic subspace clustering procedure based on *filtrations of subspace arrangements* was proposed in [19, 20]. The procedure is abstract in the sense that it receives as input a union $\mathcal{A} \subset \mathbb{R}^D$ of an unknown number of subspaces of arbitrary dimensions, and it decomposes it to the list of its constituent subspaces. This is done recursively by identifying a single subspace each time: $\mathcal{A}$ is intersected with the hyperplane $\mathcal{V}_1$, whose normal vector is the gradient of a vanishing polynomial at a point $\boldsymbol{x} \in \mathcal{A}$. Then $\mathcal{V}_1$ contains the subspace $\mathcal{S}$ associated to $\boldsymbol{x}$ and so does the new *smaller* union of subspaces $\mathcal{A}_1 = \mathcal{A} \cap \mathcal{V}_1$. Next $\mathcal{A}_1$ is intersected with a hyperplane $\mathcal{V}_2$ of $\mathcal{V}_1$, whose normal is the gradient of a vanishing polynomial of $\mathcal{A}_1$ evaluated at $\boldsymbol{x}$. As before, $\mathcal{A}_2 = \mathcal{A}_1 \cap \mathcal{V}_1$ contains $\mathcal{S}$ and the process repeats until no non-zero vanishing polynomial exists, in which case $\mathcal{S}$ is precisely $\mathcal{V}_c$, $c = D - \dim \mathcal{S}$. By picking a point $\boldsymbol{x} \in \mathcal{A} - \mathcal{S}$ a new subspace $\mathcal{S}'$ is identified and so on. This method has very strong theoretical guarantees (for noiseless data) but is fairly abstract in nature. It is the very purpose of the remaining of this paper to adapt the work of [19, 20] to a numerical algorithm and to experimentally demonstrate its merit.

3. Filtrated Spectral ASC

In this section, we propose a new subspace clustering procedure which addresses the robustness of ASC with respect to noise and unknown subspace dimensions, especially in the case of subspaces of varying dimensions.

3.1. A Distance-Based Affinity

Our first contribution is to replace the angle-based affinity in (4) by a distance-based-affinity and to show that the new affinity possesses superior theoretical guarantees.

Given unit norm data points $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^{N}$ lying close to an unknown union of n subspaces, let p be an approximate vanishing polynomial whose coefficients are given by the right singular vector of $V_n(\mathcal{X})$ associated with its smallest singular value. We define the *distance-based-affinity* as

$$ \boldsymbol{D}_{jj'} = 1 - \frac{1}{2} \big| \langle \nabla p|_{\boldsymbol{x}_j}, \boldsymbol{x}_{j'} \rangle \big| - \frac{1}{2} \big| \langle \nabla p|_{\boldsymbol{x}_{j'}}, \boldsymbol{x}_j \rangle \big| \qquad (5) $$

where the gradient vectors are assumed to be normalized to unit Euclidean norm. We will refer to this Spectral ASC

$$\mathbb{R}^D \longleftarrow \mathbb{R}^{D-1} \longleftarrow \cdots \longleftarrow \mathbb{R}^{D-c+1} \longleftarrow \mathbb{R}^{D-c} \cong \mathcal{S}$$
$$\uparrow \qquad\quad \uparrow \qquad\qquad\qquad \uparrow \qquad\qquad\quad \uparrow$$
$$\mathcal{X} \longleftarrow \bar{\mathcal{X}}_1 \longleftarrow \cdots \longleftarrow \bar{\mathcal{X}}_{c-1} \longleftarrow \bar{\mathcal{X}}_c$$

Figure 1. Commutative diagram of the filtration associated with a reference point $\boldsymbol{x} \in \mathcal{X}$. The arrows denote embeddings.

method with the *distance-based affinity* in (5) SASC-D. The denomination *distance-based* comes from the fact that the Euclidean distance from point $\boldsymbol{x}_{j'}$ to the hyperplane $\mathcal{H}^{(j)}$ defined by the unit normal vector $\nabla p|_{\boldsymbol{x}_j}$ is precisely $\left|\langle \nabla p|_{\boldsymbol{x}_j}, \boldsymbol{x}_{j'} \rangle\right|$. Moreover, $\mathcal{H}^{(j)}$ contains the subspace passing through $\boldsymbol{x}_j$. Thus, if $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ are in the same subspace, then the distance from $\boldsymbol{x}_{j'}$ to $\mathcal{H}^{(j)}$ is zero and so is the distance from $\boldsymbol{x}_j$ to $\mathcal{H}^{(j')}$. This implies that $\boldsymbol{D}_{jj'} = 1$. Of course, it may be the case that $\boldsymbol{D}_{jj'} = 1$ for points $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ coming from distinct subspaces. For instance, consider a union of two lines in $\mathbb{R}^3$ and choose a plane containing one of the lines. If the plane happens to contain the two lines, then $\boldsymbol{D}_{jj'} = 1$ for all pairs of points in the two lines.

Theorem 1 *Let $\{\boldsymbol{x}_j\}_{j=1}^N$ be points of $\mathbb{R}^D$ lying in a union of n subspaces $\mathcal{A}$. Let p be a homogeneous polynomial of any degree vanishing on $\mathcal{A}$. Then the distance-based affinity in (5) is such that if points $\boldsymbol{x}_j, \boldsymbol{x}_{j'}$ lie in the same subspace, then $D_{jj'} = 1$. The converse is not true in general.*

3.2. Filtrated ASC

Theorem 1 shows the superiority of the distance-based affinity in (5) over the angle-based affinity in (4) because it ensures that points from the same subspace will be given an affinity of maximal value 1. What still limits the theoretical guarantees of (5) is the fact that points from distinct subspaces may also have a maximal affinity 1.

In this section, we show that it is possible to further refine (5) by a filtration process illustrated in Figure 1. Let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be a set of points in $\mathbb{R}^D$ in general position in a union of n transversal subspaces $\mathcal{A} = \cup_{i=1}^n \mathcal{S}_i$. Assume that each point lies in only one of the n subspaces and is normalized to have unit norm. The key idea behind the filtration process is that, given an arbitrary *reference point* $\boldsymbol{x} \in \mathcal{X}$ in one of the subspaces, say $\mathcal{S}$, we can identify all other data points in the same subspace as $\boldsymbol{x}$ by 1) projecting all data points in $\mathcal{X}$ onto $\mathcal{S}$ and 2) finding the points in $\mathcal{X}$ whose norm after projection remains equal to one.

The fundamental challenge, however, is that we do not know $\mathcal{S}$. The filtration process in Figure 1 is designed, precisely, to perform a sequence of projections, which ultimately give the projection onto $\mathcal{S}$ without knowing $\mathcal{S}$.

At step 1 of the filtration, choose a vanishing polynomial p_0 of $\mathcal{A}$ of degree n from the nullspace of $\mathcal{V}_n(\mathcal{X})$ such that $\nabla p_0|_{\boldsymbol{x}} \neq \boldsymbol{0}$. One can show that such a p_0 always exists. Let

$\boldsymbol{b}_1 := \nabla p_0|_{\boldsymbol{x}} / ||\nabla p_0|_{\boldsymbol{x}}||$ and let $\mathcal{H}_1$ be the hyperplane of $\mathbb{R}^D$ defined by $\boldsymbol{b}_1$. If $|\langle \boldsymbol{b}_1, \boldsymbol{x}_j \rangle| > 0$, then by Theorem 1 we know that point $\boldsymbol{x}_j$ is not in $\mathcal{S}$. Consequently, we can filter the set $\mathcal{X}$ to obtain a subset $\mathcal{X}_1 := \{\boldsymbol{x}_j \in \mathcal{X} : |\langle \boldsymbol{b}_1, \boldsymbol{x}_j \rangle| = 0\}$. Geometrically, $\mathcal{X}_1$ is precisely the subset of $\mathcal{X}$ that lies inside the hyperplane $\mathcal{H}_1$, *i.e.*, $\mathcal{X}_1 = \mathcal{X} \cap \mathcal{H}_1$.

The key observation now is that $\mathcal{X}_1$ is a set of points of $\mathbb{R}^D$ drawn from the union of subspaces $\mathcal{A}_1 := \mathcal{A} \cap \mathcal{H}_1 = \bigcup_{i=1}^n (\mathcal{S}_i \cap \mathcal{H}_1)$. But $\mathcal{A}_1$ is up to isomorphism a union of subspaces of $\mathbb{R}^{D-1}$, since it is embedded in the hyperplane $\mathcal{H}_1$. In particular, consider the composite linear transformation $\pi_1 : \mathbb{R}^D \to \mathcal{H}_1 \xrightarrow{\sim} \mathbb{R}^{D-1}$, where the first arrow is the orthogonal projection of $\mathbb{R}^D$ onto $\mathcal{H}_1$ and the second arrow maps a basis of $\mathcal{H}_1$ to the standard basis of $\mathbb{R}^{D-1}$. We can replace the redundant representation $\mathcal{X}_1$ by $\bar{\mathcal{X}}_1 := \pi_1(\mathcal{X}_1) \subset \mathbb{R}^{D-1}$. It is important to note that the norm of every point in $\mathcal{X}_1$ remains unchanged and equal to 1 under the transformation π_1. Note also that now $\bar{\mathcal{X}}_1$ may be actually a subset of a union of $n_1 \leq n$ subspaces of $\mathbb{R}^{D-1}$, as it is quite possible that all points in $\mathcal{X}$ lying in some subspace $\mathcal{S}_i \neq \mathcal{S}$ were filtered out.

Now, since $\mathcal{X}$ is in general position inside $\mathcal{A}$, $\mathcal{X}_1$ will be in general position inside $\mathcal{S}_1$, and from this one can deduce that every vanishing polynomial of $\bar{\mathcal{X}}_1$ has gradient orthogonal to $\pi_1(\mathcal{S}_1)$ at $\pi(\boldsymbol{x})$, and that there is a vanishing polynomial p_1 of degree n_1 such that $\nabla p_1|_{\pi_1(\boldsymbol{x})} \neq 0$. Let $\mathcal{H}_2$ be a hyperplane of $\mathbb{R}^{D-1}$ defined by the normal vector $\boldsymbol{b}_2 := \nabla p_1|_{\pi_1(\boldsymbol{x})} / ||\nabla p_1|_{\pi_1(\boldsymbol{x})}||$. Note that $\mathcal{H}_2$ contains all the points of $\bar{\mathcal{X}}_1$ that correspond to $\mathcal{S}$. As before, we can filter the set $\bar{\mathcal{X}}_1$ to obtain a new set $\mathcal{X}_2 = \bar{\mathcal{X}}_1 \cap \mathcal{H}_2$. Once again, $\mathcal{X}_2$ lies in a union of at most n_1 subspaces of $\mathbb{R}^{D-1}$, which is however embedded in the hyperplane $\mathcal{H}_2$, and thus we can replace $\mathcal{X}_2$ by its image $\bar{\mathcal{X}}_2$ under the composite linear transformation $\pi_2 : \mathbb{R}^{D-1} \to \mathcal{H}_2 \xrightarrow{\sim} \mathbb{R}^{D-2}$, in which the first arrow is the orthogonal projection of $\mathbb{R}^{D-1}$ onto $\mathcal{H}_2$, and the second arrow maps a basis of $\mathcal{H}_2$ to the standard basis of $\mathbb{R}^{D-2}$. Proceeding inductively, this process will terminate precisely after c steps, where $c = D - \dim(\mathcal{S})$ is the codimension of $\mathcal{S}$. More specifically, there will be no non-zero vanishing polynomials on $\bar{\mathcal{X}}_c$ and $\mathcal{H}_c$ will be isomorphic to $\mathcal{S}$. Thus $\bar{\mathcal{X}}_c$ will consist of the images of the points of $\mathcal{X} \cap \mathcal{S}$ under the sequence of transformations $\pi_c \circ \pi_{c-1} \circ \cdots \circ \pi_1$. We note that the norm of these points remains unchanged and equal to 1 under $\pi_c \circ \pi_{c-1} \circ \cdots \circ \pi_1$.

Once the points of $\mathcal{X}$ that lie in $\mathcal{S}$ have been identified, we can remove them and repeat the process starting with the set $\mathcal{X} - \mathcal{X} \cap \mathcal{S}_1$, which lies in general position inside $\mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n$. This leads to Algorithm 1, which we term *Filtrated-Algebraic-Subspace-Clustering* (FASC), and is guaranteed to return the correct clustering:

Theorem 2 *Let $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^N$ be points of $\mathbb{R}^D$ lying in a transversal union of subspaces $\mathcal{A} = \cup_{i=1}^n \mathcal{S}_i$. Let $\mathcal{X}_i = \mathcal{X} \cap \mathcal{S}_i, \forall i \in [n]$. Assuming that the points of $\mathcal{X}$ are in*

Algorithm 1 Filtrated Algebraic Subspace Clustering

1: **procedure** FASC($\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^D, n$)
2:     $\mathcal{Y} \leftarrow \emptyset, \mathcal{Z} \leftarrow \emptyset$;
3:     **for** $i = 1 : n - 1$ **do**
4:         $\mathcal{X}' \leftarrow \mathcal{X} - \mathcal{Z}; d \leftarrow D$;
5:         take any $\boldsymbol{x} \in \mathcal{X}', k \leftarrow 0$;
6:         **while** $||\boldsymbol{x}|| = 1$ **do**
7:             $k \leftarrow k + 1$;
8:             find $p \in \mathcal{N}\left(\mathcal{V}_{\leq n}(\mathcal{X}')\right)$ s.t. $\nabla p|_{\boldsymbol{x}} \neq 0$;
9:             $\pi_k \leftarrow \left[\mathbb{R}^d \to \langle \nabla p|_{\boldsymbol{x}} \rangle^{\perp} \xrightarrow{\sim} \mathbb{R}^{d-1}\right]$;
10:             $\mathcal{X}' \leftarrow \{\pi_k(\boldsymbol{y}) : \boldsymbol{y} \in \mathcal{X}', \langle \nabla p|_{\boldsymbol{x}}, \boldsymbol{y} \rangle = 0\}$;
11:             $\boldsymbol{x} \leftarrow \pi_k(\boldsymbol{x}); d \leftarrow d - 1$;
12:         **end while**
13:         $\mathcal{X}' \leftarrow \{\boldsymbol{x} \in \mathcal{X} : ||\pi_{k-1} \circ \cdots \circ \pi_1(\boldsymbol{x})|| = 1\}$;
14:         $\mathcal{Z} \leftarrow \mathcal{Z} \cup \mathcal{X}'; \mathcal{Y} \leftarrow \mathcal{Y} \cup \{(\mathcal{X}', d + 1)\}$;
15:     **end for**
16:     $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{(\mathcal{X} - \mathcal{Z}, \text{rank}(\mathcal{X} - \mathcal{Z}))\}$;
17:     **return** $\mathcal{Y}$;
18: **end procedure**
---

general position inside $\mathcal{A}$, Algorithm 1 returns a set $\mathcal{Y} = \{(\mathcal{Y}_i, d_i)\}_{i=1}^{n}$ such that $\mathcal{Y}_i = \mathcal{X}_{\tau(i)}$, $d_i = \dim \mathcal{S}_{\tau(i)}$, $i = 1, \ldots, n$, where τ is a permutation on n symbols.

3.3. Filtrated Spectral ASC

Let us now consider the case where the data $\mathcal{X}$ are corrupted by noise. In this case, Algorithm 1 (FASC) is not applicable because the noisy embedded data matrix $V_n(\mathcal{X})$ is in general full rank. Nonetheless, we will show next that we can still exploit the insights revealed by the theoretical guarantees of FASC to construct a Robust-ASC algorithm.

To begin with, note that Algorithm 1 requires a single vanishing polynomial at each step of each filtration. We can use any approximate vanishing polynomial at step k. For example, letting $\bar{\mathcal{X}}_{k-1}$ be the points that have passed through the filtration at step $k - 1$, we can let p_{k-1} be the polynomial whose coefficients are given by the right singular vector of $V_n(\bar{\mathcal{X}}_{k-1})$ corresponding to its smallest singular value. Notice that no thresholding is required to choose such a p_{k-1}. This is in sharp contrast to the polynomial differentiation algorithm described in Section 2, which requires a thresholding on the singular values of $V_n(\mathcal{X})$ in order to estimate a basis of $\mathcal{I}_{\mathcal{X},n}$. Now, for any point $\boldsymbol{x} \in \mathcal{X}$, p_{k-1} gives a hyperplane $\langle \nabla p_{k-1}|_{\boldsymbol{x}} \rangle^{\perp}$ that approximately contains the subspace associated to point $\boldsymbol{x}$. However, we cannot go to the next step due to the following problems.

Problem 1 *In general, two points lying approximately in the same subspace $\mathcal{S}$ will produce different hyperplanes that approximately contain $\mathcal{S}$ with different levels of accuracy. In the noiseless case any point would be equally good. In*

the presence of noise though, the choice of the reference point $\boldsymbol{x}$ becomes significant. How should $\boldsymbol{x}$ be chosen?

Problem 2 *Given a hyperplane produced by a point $\boldsymbol{x}$, we need to determine which other points in $\mathcal{X}$ lie approximately in the hyperplane and filter out the remaining points. A simple approach is to filter out a point if its distance to the hyperplane is above a threshold δ, or if the relative change in its norm is more than δ. Clearly the choice of δ will affect the performance of the algorithm. How should δ be chosen?*

Problem 3 *Finally, we also need to determine the number of steps needed to stop the filtration. This is equivalent to determining the codimension of the subspace associated to the reference point of that filtration. In the noiseless case, one stops when the norm of the reference point becomes less than 1. In the noisy case, because the hyperplanes used to construct the filtration are only approximate, the norm of the reference point could drop at every step of the filtration. Hence a suitable stopping criterion needs to be devised.*

Inspired be the SASC-D algorithm, which handles noise by computing a normal vector for each data point and uses the normal vectors to define a distance-based affinity, we propose to address Problem 1 by constructing a filtration for each data point $\boldsymbol{x}_j \in \mathcal{X}$ with reference point $\boldsymbol{x}_j$ and using the norms of the data points to construct the affinity.

Let $\pi_k^{(j)}$ be the projection at step k of the filtration for point $\boldsymbol{x}_j$. Recall that at step k, only a subset of the original points will remain, while others will be filtered out. We can define an affinity matrix as

$$\boldsymbol{C}_{jj'}^k = \begin{cases} ||\pi_k^{(j)} \circ \cdots \circ \pi_1^{(j)}(\boldsymbol{x}_{j'})|| & \text{if } \boldsymbol{x}_{j'} \text{ remains} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

This affinity captures the fact that if points $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ are in the same subspace, then the norm of $\boldsymbol{x}_{j'}$ should not change from step 0 to step k of the filtration computed with reference point $\boldsymbol{x}_j$. Otherwise, if $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ are in different subspaces, the norm of $\boldsymbol{x}_{j'}$ is expected to be reduced by the time the filtration reaches step $c = D - \dim(\mathcal{S})$, where $\mathcal{S}$ is the reference subspace associated to $\boldsymbol{x}_j$. In the case of noiseless data, only the points in the correct subspace survive step c and their norms are precisely equal to one. Therefore, $\boldsymbol{C}_{jj'}^{(c)} = 1$ if points $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ are in the same subspace and $\boldsymbol{C}_{jj'}^{(c)} = 0$ otherwise. In the case of noisy data, the above affinity will not be perfect due to Problems 2 and 3, which we address next.

To address Problem 2, let p be the approximate vanishing polynomial whose coefficients are the right singular vector of $V_n(\mathcal{X})$ corresponding to the smallest singular value. Let

$$\beta(\mathcal{X}) = \frac{1}{N} \sum_{j=1}^{N} \left| \left\langle \boldsymbol{x}_j, \frac{\nabla p|_{\boldsymbol{x}_j}}{||\nabla p|_{\boldsymbol{x}_j}||} \right\rangle \right|. \quad (7)$$

Notice that $\beta = 0$ in the noiseless case. In the presence of noise, $\beta(\mathcal{X})$ is the average over all points of the distance of a point from the hyperplane that it produces. Evidently, small levels of noise will correspond to small values of $\beta(\mathcal{X})$. Thus, we propose to define $\delta = \gamma \cdot \beta(\mathcal{X})$, where $\gamma > 0$ is a user defined parameter. To determine γ, we propose to construct multiple filtrations for different values $\gamma_1, \ldots, \gamma_M$ of γ. Each filtration will result in a different affinity matrix. Suppose we have defined a stopping criterion to terminate each filtration so that we can use the affinity matrix at the last step of the filtration (see below for stopping criteria). Given these affinity matrices, we choose the one whose normalized Laplacian has the largest eigengap $\lambda_{n+1} - \lambda_n$, where the eigenvalues are ordered increasingly.

To address Problem 3, we stop the filtration at step k if 1) the number of points is less than the ambient dimension of the Veronese-embedded points; 2) the reference point $\boldsymbol{x}_j$ is filtered out at the $(k + 1)$th step; or 3) the number of points that passed through the filtration at step $k + 1$ is less than some integer μ. This integer is the smallest number of points that our algorithm is allowed to consider as a cluster.

Finally, the resulting affinity is symmetrized, and used for spectral clustering, as described in Algorithm 2.[1]

4. Experiments

Synthetic Data. We randomly generate $n = 3$ subspaces of dimensions $d_i \in \{1, 2, 3, 4\}$ in $\mathbb{R}^5$. For each choice of $\{d_i\}$, we randomly generate $N_i = 100$ unit norm points per subspace and add zero-mean Gaussian noise with standard deviation $\sigma \in \{0, 0.01, 0.03, 0.05\}$ in the direction orthogonal to the subspace. For each choice of $\{d_i\}$ and σ, we perform 500 independent subspace clustering experiments using the algebraic methods FSASC, SASC-D and SASC-A, and compare to state of the art methods such as SSC [6], LRR [11, 12], LRSC [23] and LSR using equation (16) in [13]. We also use the heuristic post processing of the affinity for LRR (LRR-H) and LSR (LSR-H). For FSASC we use $\mu = 10$ and $\gamma \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$, for SSC $\alpha = 20$ and $\rho = 0.7$, for LRR $\lambda = 4$, for LRSC $\tau = 420$, $\alpha = 4000$ and for LSR $\lambda = 0.0048$. We report average clustering errors, intra-cluster connectivities of the affinity matrices $\boldsymbol{C}$ produced by the methods (defined to be the minimum *algebraic connectivity*[2] among the subgraphs corresponding to each of the three subspaces) and inter-cluster connectivities ($\sum_{\boldsymbol{x}_j \in \mathcal{S}_i, \boldsymbol{x}_{j'} \in \mathcal{S}_{i'}, i \neq i'} |\boldsymbol{C}_{j,j'}| / ||\boldsymbol{C}||_1$). Due to lack of space, we report errors on all methods only

---

[1] $\text{SPECTRUM}(NL(\boldsymbol{C} + \boldsymbol{C}^\top))$ denotes the spectrum of the normalized Laplacian matrix of $\boldsymbol{C} + \boldsymbol{C}^\top$, $\text{SPECCLUST}(\boldsymbol{C}^* + (\boldsymbol{C}^*)^\top, n)$ denotes spectral clustering being applied to $\boldsymbol{C}^* + \boldsymbol{C}^{*\top}$ to obtain n clusters, and $\text{VANISHING}(V_n(\mathcal{X}))$ is the polynomial whose coefficients are the right singular vector of $V_n(\mathcal{X})$ corresponding to the smallest singular value.

[2] The algebraic connectivity of a graph is the second smallest eigenvalue of the Laplacian of the graph. Here we use the normalized Laplacian.

Algorithm 2 Filtrated Spectral ASC

1: **procedure** FSASC($\mathcal{X}, D, n, \mu, \{\gamma_m\}_{m=1}^M$)
2:     **if** $N < \mathcal{M}_n(D)$ **then**
3:         **return** ('Not enough points');
4:     **else**
5:         eigengap $\leftarrow 0$; $\boldsymbol{C}^* \leftarrow 0_{N \times N}$;
6:         $\boldsymbol{x}_j \leftarrow \boldsymbol{x}_j / ||\boldsymbol{x}_j||, \forall j \in [N]$;
7:         $p \leftarrow \text{VANISHING}(\mathcal{V}_n(\mathcal{X}))$;
8:         $\beta \leftarrow \frac{1}{N} \sum_{j=1}^N |\langle \boldsymbol{x}_j, \frac{\nabla p|_{\boldsymbol{x}_j}}{||\nabla p|_{\boldsymbol{x}_j}||} \rangle|$;
9:         **for** k = 1 : M **do**
10:            $\delta \leftarrow \beta \cdot \gamma_k$, $\boldsymbol{C} \leftarrow 0_{N \times N}$;
11:            **for** j = 1 : N **do**
12:                $C_{j,:} \leftarrow \text{FILTRATION}(\mathcal{X}, \boldsymbol{x}_j, p, \mu, \delta, n)$;
13:            **end for**
14:            $\{\lambda_s\}_{s=1}^N \leftarrow \text{SPECTRUM}(NL(\boldsymbol{C} + \boldsymbol{C}^\top))$;
15:            **if** (eigengap $< \lambda_{n+1} - \lambda_n$) **then**
16:                eigengap $\leftarrow \lambda_{n+1} - \lambda_n$; $\boldsymbol{C}^* \leftarrow \boldsymbol{C}$;
17:            **end if**
18:         **end for**
19:         $\{\mathcal{Y}_i\}_{i=1}^n \leftarrow \text{SPECCLUST}(\boldsymbol{C}^* + \boldsymbol{C}^{*\top}, n)$;
20:         **return** $\{\mathcal{Y}_i\}_{i=1}^n$;
21:     **end if**
22: **end procedure**

23: **function** FILTRATION($\mathcal{X}, \boldsymbol{x}, p, \mu, \delta, n$)
24:     $d \leftarrow D, \mathcal{J} \leftarrow [N], q \leftarrow p, \boldsymbol{c} \leftarrow 0_{1 \times N}$;
25:     flag $\leftarrow 1$;
26:     **while** ($d > 1$) and (flag $= 1$) **do**
27:         $\mathcal{H} \leftarrow \langle \nabla q|_{\boldsymbol{x}} \rangle^\perp$, $\pi \leftarrow [\mathbb{R}^d \to \mathcal{H} \xrightarrow{\sim} \mathbb{R}^{d-1}]$;
28:         **if** $(||\boldsymbol{x}|| - ||\pi(\boldsymbol{x})||)/||\boldsymbol{x}|| > \delta$ **then**
29:            **if** $d = D$ **then**
30:                $\boldsymbol{c}(j') \leftarrow ||\pi(\boldsymbol{x}'_j)||, \forall j' \in [N]$;
31:            **end if**
32:            flag $\leftarrow 0$;
33:         **else**
34:            $\mathcal{J} \leftarrow \left\{ j' \in [N] : \frac{||\boldsymbol{x}_{j'}|| - ||\pi(\boldsymbol{x}_{j'})||}{||\boldsymbol{x}_{j'}||} \leq \delta \right\}$
35:            **if** $|\mathcal{J}| < \mu$ **then**
36:                flag $\leftarrow 0$;
37:            **else**
38:                $\boldsymbol{c}(j') \leftarrow ||\pi(\boldsymbol{x}'_j)||, \forall j' \in \mathcal{J}$;
39:                $\boldsymbol{c}(j') \leftarrow 0, \forall j' \in [N] - \mathcal{J}$;
40:                **if** $|\mathcal{J}| < \mathcal{M}_n(d)$ **then**
41:                   flag $\leftarrow 0$;
42:                **else**
43:                   $d \leftarrow d - 1, \boldsymbol{x} \leftarrow \pi(\boldsymbol{x})$;
44:                   $\boldsymbol{x}_{j'} \leftarrow \pi(\boldsymbol{x}_{j'}) \forall j' \in \mathcal{J}$;
45:                   $\mathcal{X} \leftarrow \{\boldsymbol{x}_{j'} : j' \in \mathcal{J}\}$;
46:                   $q \leftarrow \text{VANISHING}(\mathcal{V}_n(\mathcal{X}))$;
47:                **end if**
48:            **end if**
49:         **end if**
50:     **end while**
51:     **return** ($\boldsymbol{c}$);
52: **end function**

---

for 0% and 5% noise.

Table 1 reports the mean clustering errors. Observe that FSASC is the only method that gives 0 error for noiseless

data for all dimension configurations, thus verifying experimentally its strong theoretical guarantees: no restrictions on the dimensions of subspaces are required for correctness. As expected, SSC, LRR, LRSC and LSR yield perfect clustering when d/D is small, but their performance degrades significantly for large d/D. Observe also that, although SASC-D is much simpler than FSASC and has similar complexity to SASC-A, its performance is very close to that of FSASC, and much better than SASC-A. We attribute this phenomenon to the correctness Theorem 1 of SASC-D. As the noise level increases, FSASC remains stable across all dimension configurations with superior behavior among all compared methods. SASC-D is less robust in the presence of noise, except for the case of hyperplanes, in which it is the best method. This phenomenon is expected, since SASC-D is essentially equivalent to FSASC if the latter is configured to take only one step in each filtration in Figure 1, and this is precisely the optimal stopping point in every filtration when the subspaces are hyperplanes. In this case, if data are noisy, the criterion for stopping FSASC filtrations is determined by the parameter γ and by the level of noise via the quantity β, leading to suboptimal values (*i.e.*, more than one step may be taken in the filtration).

Tables 2 and 3 indicate that FSASC yields higher quality affinity graphs for the purpose of clustering. To see why this is the case, observe that except for FSASC, we can distinguish two kinds of behavior in the remaining methods: the first kind gives high intra-cluster connectivity at the cost of high inter-cluster connectivity. Such methods are SASC-D, SASC-A, LRR, LRSC and LSR. The second kind gives low inter-cluster connectivity at the expense of low intra-cluster connectivity leading to unstable clustering results by the spectral clustering method. Such methods are SSC, LRR-H and LSR-H. This is expected because these methods use sparse affinities. On the other hand, FSASC circumvents this trade-off by giving high intra-cluster connectivity and low inter-cluster connectivity, thus enhancing the success of the spectral clustering step.

Motion Segmentation. We evaluate different methods on the Hopkins155 motion segmentation data set [18], which contains 155 videos of $n = 2,3$ moving objects, each one with $N = 100\text{-}500$ feature point trajectories of dimension $D = 56\text{-}80$. While SSC, LRR, LRSC and LSR can operate directly on the raw data, algebraic methods require $\mathcal{M}_n(D) \leq N$. Hence, for algebraic methods, we project the raw data onto the subspace spanned by their D principal components, where D is the largest integer ≤ 8 such that $\mathcal{M}_n(D) \leq N$, and then normalize each point to have unit norm. We apply SSC to i) the raw data (SSC-raw) and ii) the raw points projected onto their first 8 principal components and normalized to unit norm (SSC-proj). For FSASC, LRR, LRSC and LSR we use the same parameters as before, while for SSC the parameters are $\alpha = 800$ and $\rho = 0.7$.

Table 1. Mean clustering error in % over 500 independent experiments on synthetic data for 3 subspaces of $\mathbb{R}^5$ of varying dimensions (d_1, d_2, d_3) and varying levels of noise $\sigma \in \{0, 1, 3, 5\}\%$.

method	(1,1,1)	(2,2,2)	(3,3,3)	(4,4,4)	(1,2,3)	(2,3,4)
$\sigma = 0\%$						
FSASC	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
SASC-D	0.76	0.96	0.03	**0.00**	0.42	0.13
SASC-A	**0.00**	34.2	23.3	**0.00**	20.8	10.7
SSC	**0.00**	0.16	6.18	43.2	1.38	47.1
LRR	**0.00**	1.37	25.5	48.7	1.84	25.9
LRR-H	**0.00**	1.15	18.1	48.4	1.53	20.8
LRSC	**0.00**	1.32	25.6	48.7	2.27	26.5
LSR	**0.00**	7.36	34.9	48.8	5.42	28.8
LSR-H	**0.00**	1.31	28.8	29.3	1.46	32.7
$\sigma = 1\%$						
FSASC	1.70	**0.20**	**0.22**	3.17	**0.94**	**0.81**
SASC-D	2.91	0.58	0.96	**2.39**	1.78	1.89
SSC	**1.51**	1.27	6.24	42.5	8.71	44.2
$\sigma = 3\%$						
FSASC	4.39	**1.16**	**1.40**	7.67	**2.82**	**2.88**
SASC-D	8.41	2.68	4.05	**6.15**	5.40	6.46
SSC	**3.90**	5.17	7.35	44.7	9.50	33.2
$\sigma = 5\%$						
FSASC	7.02	**2.69**	**3.42**	11.34	**5.13**	**5.49**
SASC-D	13.9	6.81	8.32	**9.98**	9.98	10.9
SASC-A	43.6	47.0	32.7	12.2	41.9	27.9
SSC	**6.23**	10.2	13.2	45.9	12.1	35.4
LRR	6.74	4.88	26.2	48.8	7.70	25.5
LRR-H	6.37	3.90	20.9	48.3	6.73	23.6
LRSC	6.32	5.24	26.5	48.8	7.93	27.3
LSR	9.38	16.0	34.3	49.1	17.4	30.5
LSR-H	10.1	13.5	28.8	28.9	10.3	32.5

The clustering errors and the intra/inter-cluster connectivities are reported in Table 4 and Fig. 2. Notice the clustering errors of about 5% and 37% for SASC-A, which is the classical GPCA algorithm. Notice how changing the angle-based by the distance-based affinity (SASC-D) already gives errors of around 5.5% and 14%. But most dramatically, notice how FSASC further reduces those errors to 0.8% and 2.48%. This clearly demonstrates the advantage of FSASC over classical ASC. Moreover, even though the dimensions of the subspaces ($d_i \in \{1,2,3,4\}$ for motion segmentation) are low relative to the ambient space dimension ($D = 56\text{-}80$) - a case that is specifically suited for SSC, LRR, LRSC, LSR - projecting the data to $D \leq 8$, which makes the subspace dimensions comparable to the ambient dimension, is sufficient for FSASC to get superior performance relative to the best performing algorithms on Hopkins 155. We believe that this is because, overall, FSASC produces a much higher inter-cluster connectivity, without increasing the intra-cluster connectivity too much.

Handwritten Digit Clustering. In this section we consider the problem of clustering two digits, one of which is the digit 1 (see Benford's law, e.g. [9]). For each pair $(1, i)$, $i = 0, 2, ..., 9$, we randomly select 200 images from the MNIST database [10] corresponding to each digit and compute the clustering errors averaged over 100 independent experiments. SSC, LRR and LSR operate on raw data.

Table 2. Mean intra-cluster connectivity in % for 3 synthetic subspaces of $\mathbb{R}^5$ of dimensions (d_1, d_2, d_3) and noise σ.

method	$(1,1,1)$	$(2,2,2)$	$(3,3,3)$	$(4,4,4)$	$(1,2,3)$	$(2,3,4)$
$\sigma = 0\%$						
FSASC	100	100	100	100	100	100
SASC-D	100	100	100	100	100	100
SASC-A	100	16.6	17.7	100	18.2	16.1
SSC	40.2	0.18	0.40	0.00	0.24	0.15
LRR	100	42.0	38.1	41.6	38.4	31.6
LRR-H	100	22.2	21.5	21.5	22.9	19.8
LRSC	100	42.2	38.0	41.0	38.4	31.8
LSR	100	42.3	37.9	40.9	37.9	32.0
LSR-H	100	3.36	0.92	1.41	2.17	0.39
$\sigma = 1\%$						
FSASC	2.80	21.5	40.8	46.3	73.8	32.8
SASC-D	63.3	81.8	82.0	77.9	71.2	77.2
SSC	0.22	0.25	0.42	0.00	0.19	0.18

Table 3. Mean inter-cluster connectivity (%) for 3 synthetic subspaces of $\mathbb{R}^5$ of dimensions (d_1, d_2, d_3), and noise σ.

method	$(1,1,1)$	$(2,2,2)$	$(3,3,3)$	$(4,4,4)$	$(1,2,3)$	$(2,3,4)$
$\sigma = 0\%$						
FSASC	0.0	0.0	0.0	1.3	0.0	0.0
SASC-D	57	56	55	55	56	55
SASC-A	39	46	39	27	40	31
SSC	0.0	0.0	2.2	27	0.0	2.2
LRR	0.0	30	52	61	28	50
LRR-H	0.0	6.1	30	50	3.7	26
LRSC	0.0	31	52	61	28	51
LSR	0.0	21	38	45	19	37
LSR-H	0.0	0.0	0.0	0.0	0.0	0.0
$\sigma = 1\%$						
FSASC	8.6	4.7	11	28	9.1	20
SASC-D	60	58	58	57	59	58
SSC	0.5	0.2	2.5	26	0.5	2.9

Table 4. Mean clustering error (E), intra-cluster connectivity (C_1) and inter-cluster connectivity (C_2) in % on the Hopkins155 data.

method	2 motions			3 motions			all motions		
	E	C_1	C_2	E	C_1	C_2	E	C_1	C_2
FSASC	**0.80**	18	4	**2.48**	10	10	**1.18**	16	5
SASC-D	5.65	82	26	14.0	80	46	7.59	81	31
SASC-A	4.99	35	5	36.8	9	35	12.2	29	12
SSC-raw	1.53	5	2	4.40	4	3	2.18	5	2
SSC-proj	5.87	4	3	5.70	3	3	5.83	3	3
LRR	4.26	25	19	7.78	25	28	5.05	25	21
LRR-H	2.25	5	2	3.40	4	3	2.51	5	2
LRSC	3.38	25	19	7.42	24	28	4.29	25	21
LSR	3.60	24	18	7.77	23	28	4.54	23	21
LSR-H	2.73	4	1	2.60	3	2	2.70	4	1

LRR and LSR parameters are the same as before. For FSASC we set $\mu = 10$ and $\gamma = 1$ and for SSC we set $\alpha = 10$ and $\rho = 0.7$, as before. For the three algebraic methods we first project the raw data onto their first 13 principal components and then normalize each point to have unit norm. For comparison, we also run SSC on the projected data. Mean errors are reported in Table 5 with SASC-A, LRR, LRSC omitted since they perform poorly (with LRR performing worse with the post-processing). We also do not show the numbers for SSC-proj since they are very
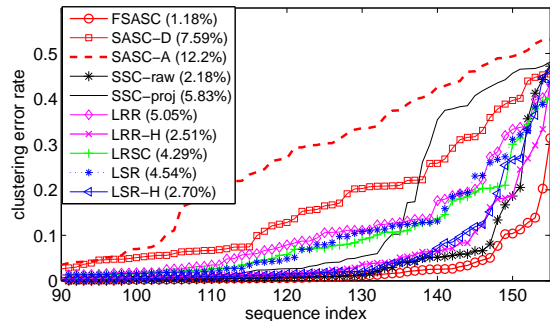

Figure 2. Clustering error ratios for both 2 and 3 motions in Hopkins155, ordered increasingly for each method. Errors start from the 90-th smallest error of each method.

Table 5. Clustering error (%) for two digits $(1, i)$, $i = 0, 2, ..., 9$ in the MNIST dataset.

method	Digits Pair								
	$(1,0)$	$(1,2)$	$(1,3)$	$(1,4)$	$(1,5)$	$(1,6)$	$(1,7)$	$(1,8)$	$(1,9)$
FSASC	**0.50**	4.67	1.55	3.31	1.11	1.62	2.27	4.88	1.81
SASC-D	4.91	14.2	10.3	23.9	8.55	13.1	10.2	21.5	17.5
SSC-raw	1.12	9.15	2.66	5.77	2.78	1.87	2.90	13.3	2.00
LSR	1.03	5.26	2.13	19.1	1.29	1.50	5.40	15.3	5.90
LSR-H	0.74	**1.35**	**1.12**	**3.15**	**0.88**	**0.90**	**1.33**	**4.38**	**1.11**

close to those of SSC-raw. As in the case of motion segmentation, we observe that FSASC outperforms SASC-D (this time by a large margin), which in turn significantly outperforms SASC-A. This confirms the superiority of FSASC over previous algebraic methods. As before, FSASC is also superior to SSC. The only method that performs better is LSR-H. We note that projecting the 784-dimensional data onto dimension 13, reduces the angles between the subspaces, thus making the clustering problem harder. As a result, for more than 2 digits the performance of FSASC degrades singinficantly, even for projection dimensions up to $D = 17$, since it becomes harder for the method to distinguish the subspaces. To circumvent this issue, a higher projection dimension would be required, which currently can not be handled by FSASC, due to the high complexity.

5. Conclusions

We presented a novel algebraic subspace clustering method based on the geometric idea of filtrations and we experimentally demonstrated its robustness to noise using synthetic and real data and its superiority to the state-of-the-art algorithms on several occasions. Overall, the method works very well for subspaces of arbitrary dimensions in a low-dimensional ambient space, and it can handle higher dimensions via a projection. The main weakness of the method is its high computational complexity, which comes from the large number of filtrations required, as well as the exponential cost of fitting polynomials to n subspaces. Future research will be concerned with reducing the complexity, as well as dealing with outliers and missing entries.

References

[1] M. Atiyah and I. MacDonald. *Introduction to Commutative Algebra*. Westview Press, 1994.

[2] P. S. Bradley and O. L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[3] H. Derksen. Hilbert series of subspace arrangements. *Journal of Pure and Applied Algebra*, 209(1):91–98, 2007.

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[5] E. Elhamifar and R. Vidal. Clustering disjoint subspaces via sparse representation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[6] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[7] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[8] K. Huang, Y. Ma, and R. Vidal. Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 631–638, 2004.

[9] A. Kossovsky. *Benford's Law : Theory, the General Law of Relative Quantities, and Forensic Fraud Detection Applications*. World Scientific Publishing Company, 2014.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278 – 2324, 1998.

[11] G. Liu, Z. Lin, S. Yan, J. Sun, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(1):171–184, Jan 2013.

[12] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010.

[13] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *European Conference on Computer Vision*, 2012.

[14] Y. Ma, A. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008.

[15] B. McWilliams and G. Montana. Subspace clustering of high-dimensional data: a predictive approach. *Data Mining and Knowledge Discovery*, 28(3):736–772, 2014.

[16] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2012.

[17] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[18] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[19] M. C. Tsakiris and R. Vidal. Abstract algebraic-geometric subspace clustering. In *Proceedings of Asilomar Conference on Signals, Systems and Computers*, 2014.

[20] M. C. Tsakiris and R. Vidal. Abstract algebraic subspace clustering. *CoRR*, abs/1506.06289, 2015.

[21] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000.

[22] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(3):52–68, March 2011.

[23] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43:47–61, 2014.

[24] R. Vidal, Y. Ma, and J. Piazzi. A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 510–517, 2004.

[25] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 621–628, 2003.

[26] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005.